UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/849,736	05/04/2001	G. Glenn Henry	CNTR:2021	8084

23669 7590 07/11/2005

HUFFMAN LAW GROUP, P.C.
1832 N. CASCADE AVE.
COLORADO SPRINGS, CO 80907-7449

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 07/11/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/849,736	HENRY ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 April 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *31,34-38 and 43-65* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *31,34-38 and 43-65* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *04 Amy 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 31, 34-38, and 43-65 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: RCE and Amendment as received on 4/25/2005.

Information Disclosure Statement

3. The examiner stated in the previous Office Action that the NPL portion of the IDS filed on 12/15/2004 was not considered. However, the references were apparently considered since they were initialed, so the statement in the previous Office Action about them not being considered appears to be an error.

Withdrawn Rejections

4. Applicant has overcome, via amendment, the prior art rejections set forth in the previous Office Action. Consequently, these rejections are hereby withdrawn by the examiner. However, upon further consideration, a new grounds of rejection is applied below.

Claim Objections

5. Claim 1 is objected to because of the following informalities: In line 4, replace "comprises" with --comprise--. Appropriate correction is required.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claim 31 recites the limitation "said instruction" twice in line 13. There is insufficient antecedent basis for this limitation in the claim, as applicant previously claims "said instructions".

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 31, 43-44, and 46-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Narayan et al., U.S. Patent No. 5,850,532 (herein referred to as Narayan) in view of Hoyt et al., U.S. Patent No. 5,604,877 (herein referred to as Hoyt).

10. Referring to claim 31, Narayan has taught a pipelined microprocessor comprising:

a) an instruction cache (Fig.1, component 16) that is indexed by a fetch address, said instruction cache for caching instructions, wherein said instructions comprise variable byte-length instructions. See column 4, lines 36-37.

b) Narayan has not explicitly taught providing said instructions to an instruction buffer for storage therein. However, Official Notice is taken that an instruction buffer, such as a prefetch buffer, is well known and accepted in the art. More specifically, a prefetch buffer prefetches and

stores instructions from the instruction cache, thereby making instructions available to the processor immediately, as opposed to the processor having to perform a somewhat time-expensive fetch from cache memory. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Narayan to include a buffer which receives and stores instructions from the instruction cache.

c) a branch target address cache (Fig. 1, component 14), coupled to said instruction buffer and indexed by said fetch address, for caching branch target addresses of previously executed branch instructions. See column 7, lines 43-46.

d) said instruction buffer comprising an indicator associated with each byte of each of said instructions stored in said instruction buffer, wherein said indicator has a true value if said branch target address cache predicts that said byte is an opcode byte. See column 2, lines 41-43, and note a functional bit's setting determines whether the corresponding byte is an opcode byte.

e) Narayan has not taught that the indicator also indicates that said instruction is one of said previously executed branch instructions and the microprocessor has speculatively branched to one of said branch target addresses cached for said one of said previously executed branch instructions. However, Hoyt has taught such a concept. See column 12, lines 58-64. Note that an indication is used to indicate that a prediction was made for a branch. Hoyt implements such an indication to track which instructions the system believes are branched. The belief is then verified (or not verified) by additional logic. See column 13, lines 7-10. Such a system allows for early branch prediction, before the branch is decoded so that instructions may continue to be fetched immediately. As a result, in order to fetch instructions along a predicted path sooner, it

would have been obvious to one of ordinary skill in the art at the time of the invention to modify Narayan to have such an indication.

11. Referring to claim 43, Narayan in view of Hoyt has taught a microprocessor as described in claim 31.

a) Narayan has further taught instruction decode logic, coupled to said instruction buffer, for decoding said instructions to indicate which byte of each of said instructions is an opcode byte. See Fig.1, component 12.

b) Narayan in view of Hoyt has not taught prediction check logic, coupled to receive said indicator associated with each byte of said instruction from said instruction buffer, wherein if one of said indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said byte of said one of said instructions associated with said one of said indicators is not indicated by said instruction decode logic to be said opcode byte. However, Official Notice is taken that it is common for caches (BTAC/BTB included) to be constructed such that multiple fetch addresses map to the same cache entry (technique referred to as aliasing). If in the situation where a non-opcode byte is encountered and it maps to a branch entry that is allocated to a branch opcode, then a prediction should not be made because it is unknown whether this non-opcode byte is actually associated with a branch instruction. But, since it maps to the same location as a branch, the system will assume a branch has been encountered and an erroneous prediction would have been made. Such a cache construction is desired because it is not often that instructions map to the same location and it

also keeps the cache smaller. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to reduce the size of the BTAC and determine erroneous predictions if a non-opcode byte results in a prediction being made.

12. Referring to claim 44, Narayan in view of Hoyt has taught a microprocessor as described in claim 31.

a) Narayan has further taught instruction decode logic, coupled to said instruction buffer, for decoding said instructions to indicate which byte of each of said instructions is an opcode byte. See Fig.1, component 12.

b) Hoyt has further taught prediction check logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein if one of said indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said instruction decode logic indicates said one of said instructions is a non-branch instruction. See column 13, lines 7-19.

13. Referring to claim 46, Narayan in view of Hoyt has taught a microprocessor as described in claim 31. Furthermore, Narayan in view of Hoyt has taught prediction check logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein said one of said instructions is a branch instruction, wherein if one of said indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target

addresses if a resolved direction of said branch instruction does not match a direction of said branch instruction predicted by said branch target address cache. This is deemed to be inherent because if the resolved direction of a branch does not match the predicted direction, then a misprediction has occurred (erroneously branched).

14. Referring to claim 47, Narayan in view of Hoyt has taught a microprocessor as described in claim 31. Narayan in view of Hoyt has further taught prediction check logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein said one of said instructions is a branch instruction, wherein if one of said indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if a resolved target address of said branch instruction does not match said one of said branch target addresses to which the microprocessor speculatively branched. This is deemed to be inherent because if the resolved target address of a branch does not match the speculative target address, then a misprediction has occurred (erroneously branched).

15. Referring to claim 48, Narayan in view of Hoyt has taught a microprocessor as described in claim 31. Hoyt has further taught a non-speculative branch predictor (Fig.5, component 51), coupled to said instruction buffer, for generating a non-speculative predicted target address of a branch instruction for which said branch target address cache provided said one of said branch target addresses to which the microprocessor speculatively branched, and branch control logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein if one of said indicators associated with said branch instruction

specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said branch control logic causes the microprocessor to branch to said non-speculative predicted target address if said non-speculative predicted target address generated by said non-speculative branch predictor does not match said one of said branch target addresses of said branch instruction provided by said branch target address cache. See column 13, lines 45-67.

16. Referring to claim 49, Narayan in view of Hoyt has taught a microprocessor as described in claim 31. Hoyt has further taught a non-speculative branch predictor, coupled to said instruction buffer, for generating a non-speculative predicted direction of a branch instruction for which said branch target address cache provided said one of said branch target addresses to which the microprocessor speculatively branched, and branch control logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein if one of said indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said branch control logic causes the microprocessor to branch to a next instruction sequential to said branch instruction if said non-speculative predicted direction generated by said non-speculative branch predictor is a not taken prediction. See column 13, lines 45-67.

17. Claims 34-36, 50-55, 59, and 61-65 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell et al., U.S. Patent No. 5,850,543 (as applied in the previous Office Action and herein referred to as Shiell) in view of Roberts et al., U.S. Patent No. 5,752,069 (herein referred to as Roberts).

18. Referring to claim 34, Shiell has taught a method of speculatively branching in a pipelined microprocessor, comprising:

a) caching, in a branch target address cache (BTAC), a plurality of branch target addresses of previously executed branch instructions. See Fig.2, BTB 56 and column 7, lines 40-42.

b) Shiell has not taught caching a bit associated with each of said branch instructions, wherein said bit is true only if the associated branch instruction spans more than one instruction cache line. However, Roberts has taught a single bit associated with each instruction which is only true if the instructions spans multiple cache lines. See bit 1 of the ICMROM signal (column 30, lines 27-30). Such an indication informs the system that it must fetch the remaining portion of an instruction from the next cache line. By allowing instructions to span multiple lines, cache storage would be utilized more efficiently. For instance, if each cache line holds 4 bytes, and a subset of instructions are 3 bytes in length, then if there is no wrapping, one byte of each line would be wasted (since you can't fit a second instruction in a line). However, with wrapping, that final byte of storage is utilized every time. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Shiell to cache an indication of whether an instruction spans an instruction cache line.

c) accessing said BTAC with a fetch address of an instruction cache after said caching (fig. 2 indicates that the fetch address FA is used to access the BTB 56; col. 8, lines 13-16 indicate that the accessing is done when branch history is stored in the BTB).

d) determining whether said fetch address hits in said BTAC in response to said accessing (col. 8, lines 13-16).

e) branching the microprocessor to one of said plurality of branch target addresses selected by said fetch address if said fetch address hits in said BTAC whether or not a branch instruction is cached in a line of said instruction cache indexed by said fetch address (col. 7, lines 40-45 indicate that the target address is used to generate the fetch address during the fetch stage before the decoding the instruction without knowledge of whether the instruction at that fetch address in the instruction cache is still a branch or not i.e. speculative execution [col. 8, lines 13-16] because the BTB and the instruction cache are accessed in parallel using the fetch address FA).

19. Referring to claim 35, Shiell in view of Roberts has taught a method as described in claim 34. Shiell has further taught storing a branch direction prediction associated with each of said plurality of branch target addresses prior to said accessing said BTAC (fig. 3, "HIS_n" field; col. 8, lines 57-67).

20. Referring to claim 36, Shiell in view of Roberts has taught a method as described in claim 35. Shiell has further taught that said branching the microprocessor to said one of said plurality of branch target addresses selected by said fetch address is performed only if said associated branch direction prediction indicates said branch instruction will be taken (Although not explicitly mentioned, the limitation is deemed inherent to the correct functioning of the method because the purpose of the prediction information when indicating that the speculative branch is taken is for instructing the processor to branch to the target address of the branch and not the next sequential address).

21. Referring to claim 50, Shiell has taught a branch target address cache (BTAC) (fig. 2, Branch Target Buffer, BTB 56) for providing a speculative target address (col. 7, lines 40-42; the target address is speculative because it is provided during the instruction fetch stage before it is

known whether the instruction is a branch or not) to address selection logic (fig. 2, multiplexers 57, 58 and 52), the address selection logic selecting a fetch address for addressing a line in an instruction cache (fig. 2 shows that the address selection logic addresses the instruction cache 16_i and col. 6, lines 16-23 indicate that the fetch address addresses a stream of instruction data i.e. a line in the instruction cache), the BTAC providing the speculative target address based on a presumption that a branch instruction is present in the cache (fig. 2 shows that the BTB outputs a target address D0-D127 indexed by the fetch address without knowing whether or not a branch is present in the line of the instruction cache addressed by the fetch address because the BTB and the instruction cache are accessed in parallel using the fetch address FA), the BTAC comprising:

a) an array of storage elements, configured to cache target addresses of previously executed branch instructions (fig. 2, BTB 56; col. 7, lines 40-42).

b) Shiell has not taught said speculative branch information comprises a bit associated with each of said branch instructions, wherein said bit is true only if the associated branch instruction presumed present in the cache line spans more than one line in the instruction cache. However, Roberts has taught a single bit associated with each instruction which is only true if the instructions spans multiple cache lines. See bit 1 of the ICMROM signal (column 30, lines 27-30). Such an indication informs the system that it must fetch the remaining portion of an instruction from the next cache line. By allowing instructions to span multiple lines, cache storage would be utilized more efficiently. For instance, if each cache line holds 4 bytes, and a subset of instructions are 3 bytes in length, then if there is no wrapping, one byte of each line would be wasted (since you can't fit a second instruction in a line). However, with wrapping, that final byte of storage is utilized every time. Consequently, it would have been obvious to one

of ordinary skill in the art at the time of the invention to modify Shiell to cache an indication of whether an instruction spans an instruction cache line.

c) an input, coupled to said array, for receiving the fetch address, to index into said array of storage elements to select one of said target addresses (fig. 2; col. 8, lines 12-16).

d) an output, coupled to said array, for providing said one of said target addresses indexed by the fetch address to the address selection logic (fig. 2 shows the target address D0-D127 are connected to an output to the address selection logic [multiplexers 57, 58 and 52]);

e) wherein said output provides said one of said target addresses to the address selection logic for selection as a subsequent fetch address whether or not a branch instruction is present in the line of the instruction cache addressed by the fetch address (fig. 2 shows that the BTB outputs a target address D0-D127 indexed by the fetch address to the address selection logic (multiplexers 57, 58 and 52) without knowing whether or not a branch is present in the line of the instruction cache addressed by the fetch address because the BTB and the instruction cache are accessed in parallel using the fetch address FA).

22. Referring to claim 51, Shiell in view of Roberts has taught a BTAC as described in claim 50. Shiell has further taught a second output, coupled to said array, for providing a portion of said speculative branch information to control logic for controlling the address selection logic in response to said portion of said speculative branch information (although not shown, it is deemed inherent to the BTB to have a second output which outputs a portion of the history information indicating the outcome of the branch to control the address selection logic multiplexers. This is because when the branch is indicated in the BTB as not taken, the BTB target address should not be selected and the next sequential address should be selected).

23. Referring to claim 52, Shiell in view of Roberts has taught a BTAC as described in claim 50. Shiell has further taught that said speculative branch information comprises information predicting whether the branch instruction presumed present in the cache line will be taken (fig. 3, "HIS" field indicates that the branch will be taken if set to '111' or '110' [col. 8, lines 57-67]).

24. Referring to claim 53, Shiell in view of Roberts has taught a BTAC as described in claim 52. Shiell has further taught that said information predicting whether the presumed branch instruction will be taken comprises a taken/not taken bit (fig. 9, when a conditional branch, the 2nd bit in the "HIS" field is a taken/not taken bit because when set to "1" it indicates taken, when set to "0" it indicates not taken).

25. Referring to claim 54, Shiell in view of Roberts has taught a BTAC as described in claim 52. Shiell has further taught that said information predicting whether the presumed branch instruction will be taken comprises a plurality of bits (fig. 3, "HIS" field).

26. Referring to claim 55, Shiell in view of Roberts has taught a BTAC as described in claim 54. Shiell has further taught that said plurality of bits is stored in a saturating up/down counter (col. 9, lines 5-18).

27. Referring to claim 59, Shiell in view of Roberts has taught a BTAC as described in claim 50. Shiell has further taught that said speculative branch information comprises information specifying a location within the cache line of the branch instruction presumed present in the cache line (fig. 3, shows the "T_n" field as an entry in the BTB. col. 8, lines 40-44 indicate that the T_n field holds information specifying the location of a specific instruction within the cache line associated with the logical address LA used to index into the BTB).

28. Referring to claim 61, Shiell in view of Roberts has taught a BTAC as described in claim 50. Shiell has further taught that said speculative branch information comprises an indication of a type of the branch instruction presumed present in the cache line (fig. 2, "HIS" field, col. 8, lines 57-67).

29. Referring to claim 62, Shiell in view of Roberts has taught a BTAC as described in claim 61. Shiell has further taught that said indication of said type of the branch instruction indicates whether the branch instruction is a call instruction ("HIS" = 011; col. 8, lines 57-67).

30. Referring to claim 63, Shiell in view of Roberts has taught a BTAC as described in claim 61. Shiell has further taught that said indication of said type of the branch instruction indicates whether the branch instruction is a return instruction ("HIS" = 010; col. 8, lines 57-67).

31. Referring to claim 64, Shiell in view of Roberts has taught a BTAC as described in claim 50. Shiell has further taught that each of said storage elements is configured to cache a plurality of target addresses (col. 8, lines 22-27 indicates that there are 4 target addresses stored per storage element).

32. Referring to claim 65, Shiell in view of Roberts has taught a BTAC as described in claim 50. Shiell has further taught that the branch target address cache is external to the instruction cache. See Fig.2 and note that the BTAC 56 is within fetch unit 26, which is shown in Fig.1 as being external to the I-cache.

33. Claims 37-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell in view of Roberts, as applied above, and further in view of Narayan and Hoyt, as applied above.

34. Referring to claim 37, Shiell in view of Roberts has taught a method as described in claim 34.

a) Shiell has further taught that the microprocessor processes variable byte-length instructions. See column 6, lines 32-34.

b) Shiell has further taught storing in an instruction buffer instructions provided by said instruction cache selected by said fetch address. See Fig.2, component 60.

c) Shiell has not taught storing a discrete indication for each byte of said instruction, wherein said discrete indication is true if said BTAC predicts said byte is an opcode byte of said instruction. However, Narayan has taught indicating an opcode byte. See column 2, lines 41-43, and note a functional bit's setting determines whether the corresponding byte is an opcode byte. Clearly, the system should have knowledge of where the opcode of an instruction is, otherwise, it would not be able to determine the type of instruction. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Shiell such that Shiell includes an indication as to the location of opcode bytes.

d) Shiell and Narayan have not taught that the indication is true if said branching the microprocessor to one of said plurality of branch target addresses was performed for said instruction. However, Hoyt has taught such a concept. See column 12, lines 58-64. Note that an indication is used to indicate that a prediction was made for a branch. Hoyt implements such an indication to track which instructions the system believes are branched. The belief is then verified (or not verified) by additional logic. See column 13, lines 7-10. Such a system allows for early branch prediction, before the branch is decoded so that instructions may continue to be fetched immediately. As a result, in order to fetch instructions along a predicted path sooner, it

would have been obvious to one of ordinary skill in the art at the time of the invention to modify Shiell to have such an indication.

35. Referring to claim 38, Shiell in view of Roberts in view of Narayan and further in view of Hoyt has taught a method as described in claim 39. Hoyt has further taught determining from said discrete indication, subsequent to said storing, that said branching was performed. See column 13, lines 16-19.

36. Claim 45 is rejected under 35 U.S.C. 103(a) as being unpatentable over Narayan in view of Hoyt, as applied above, and further in view of Stiles, U.S. Patent No. 5,513,330 (as applied in the previous Office Action).

37. Referring to claim 45, Narayan in view of Hoyt has taught a microprocessor as described in claim 31.

a) Narayan has further taught instruction decode logic, coupled to said instruction buffer. See Fig.1.

b) Black has not taught that the BTAC is configured to cache a length of each of said previously executed branch instructions, the decode logic determines a length of each of said instructions, and prediction check logic, coupled to receive said indicator associated with each byte of said instructions from said instruction buffer, wherein if one of said indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said previously executed branch instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said length received from said instruction decode logic does not match

said length of said one of said previously executed branch instructions provided by said branch target address cache. However, Stiles has taught a branch prediction cache which supplies target instruction lengths so that alignment and fetching may be done efficiently. See the first paragraph in the "Summary of Invention" section. Stiles has also taught calculating the lengths of target instructions which are not yet registered in the branch cache. See Fig.2 and Fig.4. Although Stiles has not taught comparing the predicted and calculated addresses, a person of ordinary skill in the art would have recognized that the comparison would be useful in systems which experience aliasing (where multiple branches map to the same entry in cache). Basically, aliasing is sometimes allowed so that the cache may be kept smaller (there will not be an individual entry for each instruction). The hope is that it isn't often that instructions map to the same location, and so the reduction in cache size is worth the occasional penalty of an instruction mapping to a shared location. As a result it would have been obvious to one of ordinary skill in the art to implement aliasing in Narayan in view of Hoyt. However, with aliasing implemented, a branch might map to an address which belongs to another branch, and consequently, the target instructions and their lengths may be incorrect. Therefore, a misprediction can be determined by comparing the predicted addresses and the calculated addresses for the actual instructions. Therefore, in order to detect mispredictions in a system with aliasing, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Narayan in view of Hoyt to include a comparison of predicted and actual instruction lengths.

38. Claims 56-58 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell in view of Roberts, as applied above, and further in view of Bae et al., U.S. Patent No. 6,044,459 (as applied in the previous Office Action and herein referred to as Bae).

39. Referring to claim 56, Shiell in view of Roberts has taught a BTAC as described in claim 51. Furthermore, although Shiell verifies whether the target address was correct by comparing it with the actual next instruction address determined by the execution unit (col. 2, lines 48-51), Shiell has not taught that said portion of said speculative branch information comprises an indication of whether said one of said target addresses is a valid target address. However, Bae has taught a BTB entry format with a valid bit for indicating whether the target address is a valid target address (col. 5, lines 27-34). The valid bit is checked before providing the target address as the fetch address so that an invalid target is not incorrectly provided. It would have been obvious to one of ordinary skill in the art at the time of the invention to recognize to add a valid bit in the BTB of Shiell indicating the validity of the target address so that in invalid target is not provided. One would have been motivated to do so because by including the valid bit, incorrect targets are not fetched and subsequent flushing of the pipeline is not required, thereby leading to improved performance.

40. Referring to claim 57, Shiell in view of Roberts has taught a BTAC as described in claim 56. Shiell in view of Roberts and further in view of Bae has further taught that said indication is populated to indicate said one of said target addresses is a valid target address in response to execution of the presumed branch instruction, wherein said one of said target addresses is resolved. This limitation is deemed inherent because the valid bit can be set to indicate the

validity of the predicted target address only after the branch instruction is executed and the target is calculated.

41. Referring to claim 58, Shiell in view of Roberts has taught a BTAC as described in claim 56. Shiell in view of Roberts and further in view of Bae has further taught that said indication is populated to indicate said one of said target addresses is not a valid target address in response to detecting said one of said target addresses is erroneous subsequent to said providing said one of said target addresses on said output (This limitation is deemed inherent because the valid bit can be set to indicate that the predicted target address is invalid only after comparing the predicted target address in the BTB with the target address calculated on execution i.e. detecting that the target address is erroneous).

42. Claim 60 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell in view of Roberts, as applied above, and further in view of Hsu et al., U.S. Patent No. 5,948,100 (as applied in the previous Office Action and herein referred to as Hsu).

43. Referring to claim 60, Shiell in view of Roberts has taught a BTAC as described in claim 50. Shiell has not taught that said speculative branch information comprises a length of the branch instruction presumed present in the cache line. However, Hsu has taught a BTB which caches the length of each branch instruction. See column 13, lines 18-30. This allows the system to accommodate branch instructions that cross multiple cache blocks, which is useful in variable length instruction systems, as both Hsu and Shiell are (see the title of Hsu and column 6, lines 32-35 of Shiell). Consequently, it would have been obvious to one of ordinary skill in the

art at the time of the invention to modify Shiell to cache the length of a branch instruction so that the system may accommodate branches that span multiple cache lines.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
June 29, 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100